

ハミングによる検索機能を備えた 音楽配信システムの開発

獅々堀 正幹¹, 柘植 寛¹, 北 研二²

Development of Similar Music Retrieval Systems for the Query-by-Humming

by

Masami SHISHIBORI, Satoru TSUGE, Kenji KITA

Abstract

Music retrieval systems are extremely useful for collecting digital music data from on-line music distribution sites. Especially, there is a great need to develop effective techniques for content-based music retrieval systems, which can retrieve by humming query. The main issues in this research is how to decide the similarity of each music features extracted from music data. In order to calculate the similarity, some conventional methods use Euclid distance or DP matching, but it is very hard to solve the problem of the vagueness of humming query. In this paper, we propose a new similar music retrieval method based on humming query using the Earth Mover's Distance as the distance measure. Computing the EMD is based on a solution to the transportation problem, and the EMD is applied as the distance measure on similar image retrieval systems. In addition, we focus that the time complexity of the EMD is exponential worst case toward the number of notes, the improved method to decrease the number of notes in the music feature is also proposed. Experimental results show that the proposed method can improve the retrieval precision of conventional systems.

Key words: Information Retrieval, Query-by-Humming, Earth Mover's Distance

1. はじめに

近年、マルチメディアデータの圧縮技術の向上、及びインターネットの普及により、楽曲配信サービスを行うサイトも多数出現している。それに伴い、ユーザは容易に音楽データをパソコンや携帯用音楽プレイヤーにダウンロードして試聴できるようになった。しかし、サーバ側で蓄積される音楽データが膨大になるにつれ、ユーザは目的の曲を見つけ出すことが困難となり、音楽データに対する効率の良い検索手法が必要になっている⁽¹⁾。

音楽データに対する検索方式は、曲名、歌手名や歌詞等を入力とするキーワード型の検索方式が一般的であるが、より汎用的なコンテンツ型の検索方式が注目されている⁽²⁾。コンテンツ型検索方式は、ユーザが曲の一部のリズムを入力するシステム⁽³⁻⁴⁾、実際にハミングを入力するシステム⁽⁵⁻¹⁷⁾、感性情報により検索するシステム⁽¹⁸⁾、ユーザの嗜好に基づいて検索するシステム⁽¹⁹⁾等が存在する。特にハミング検索システムは、曲名や歌手名を記憶していなくても、メロディだけを覚えていれば検索できるため、発展が期待されている⁽²⁾。このような背景から、我々はコンテンツ型検索方式の中でも特にハミングによりMIDI形式の音楽データを検索する研究を行っている。

従来のハミング検索システムとしては、量子化した音楽データに対してn-gram検索を行う手法⁽⁵⁾、ハミングデータ内に発生する誤りパターンをHMMで学習し、確率的に類似性を判定する手法⁽⁶⁻⁷⁾、音楽データの音長や音高の変化の類似性をDPマッチングを用いて判定する手法、音高の変化を固定長のベクトルで表現し、ベクトル間の類似性をユークリッド距離で判定する手法⁽⁸⁻¹⁰⁾等が存

1 徳島大学大学院ソシオテクノサイエンス研究部
情報ソリューション部門

Information Solution, Institute of Technology and
Science, The University of Tokushima

2 徳島大学高度情報化基盤センター

Center for Advanced Information Technology,

The University of Tokushima

連絡先：〒770-8506 徳島市南常三島町2-1

在する．このように様々な距離尺度を用いた手法が提案されているが，音長と音高等の様々な音楽特徴量の類似性を同じ範疇の距離尺度で判定することが重要である．文献(12,20)では，音長と音高の距離の総和を用いてDPマッチングを行うことで，単独の特徴量を用いた場合よりも大幅な精度向上に成功している．

本論文では，距離尺度としてEarth Mover's Distance (EMD)⁽²¹⁾を用いたハミングによる類似音楽検索手法を提案する．EMDは線形計画問題の一つである輸送問題における輸送コストの最適解を求めるアルゴリズムである．また，Rubnerら⁽²¹⁾により類似画像検索の距離尺度に適用され，検索精度の向上に成功している．EMDを用いた類似画像検索では，画像を色情報により領域分割し，各領域を供給地，各領域の広さを供給地が有する資源量とみなし，各領域の重心点，平均色等の特徴量から輸送コストを定義することで，画像全体の構図が類似した（同じ位置に類似した色が配色された）画像を検索できるといった長所を持つ．

本手法では，音楽データを固定長のメロディ片に分割した後，メロディ片内の各音符を供給地，各音符の音長を供給地の資源量とみなし，各音符の出現時間，音高等の特徴量から輸送コストを算出する．この音楽特徴量をEMDで距離計算することで，リズムと音程との類似度を同じ距離尺度で計り，全体の曲調が類似したメロディを検索する．一方，供給地数を N とすると，EMDの計算量は N に対して指数関数的に増加する．即ち，EMDを音楽特徴量に適用した際には，計算量が音符数に対して指数関数的に増加する．そこで本システムでは，EMDの下限値を用いた高速検索アルゴリズムを適用し，大規模なデータベースに対しても高速に検索結果を得ることに成功した．

以下，2章でハミングによる類似音楽検索システムの概要を述べる．3章ではハミングをMIDI音楽データに変化する採譜手法について述べる．4章では距離尺度にEMDを用いたハミング検索手法を提案する．5章において実際のハミングデータを用いた実験結果を示し，考察を述べる．最後に6章において，まとめ及び今後の課題について述べる．

2. ハミングによる類似音楽検索システム

ハミングによる類似音楽検索システムでは，図1に示すように，ユーザは検索したい曲の一部をハミングする．そして，ハミングデータと類似した曲が音楽データベースから検索され，順位付けされた結果がユーザに提示される．

本システムでは，まず，前処理として検索対象のMIDI形式の音楽データから主旋律を抽出し，主旋律をスライディング・ウィンドウ方式⁽¹⁹⁾によりメロディ片に分割する．スライディング・ウィンドウ方式とは，図1に示すように，固定ウィンドウ長に対して「拍」を単位に，ある幅ずつスライドしてメロディ片に分割する方法である．スライド幅をウィンドウ長より短くすることで，連続するメロディ片は互いに重なりのある冗長なデータとなり，検索する部分に関して自由度が増すことになる．その後，メロディ片毎に音長や音高といった音楽特徴量を抽出し，音楽特徴量データベースを作成する．

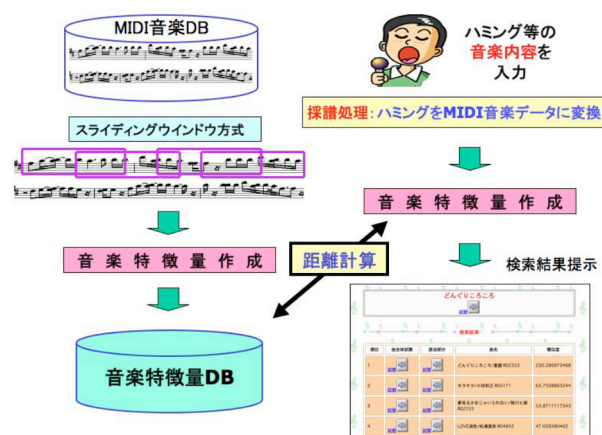


Fig.1 Similar music retrieval systems for the query-by-humming.

次に，ユーザからハミングされたデータをMIDI形式の音楽データに変換する．この処理を採譜処理と呼ぶ．採譜されたMIDI形式のハミングデータも同様にスライディング・ウィンドウ方式により，メロディ片に分割した後，音楽特徴量を抽出する．その後，入力ハミングの音楽特徴量と音楽特徴量データベース内の距離計算を行い，距離の近い類似したメロディを持つ曲を検索する．検索キーがハミングになるため，音程のずれやリズムの違いの吸収等が重要な課題になる．

従来のハミング検索手法としては，DPマッチングを用いる手法，ユークリッド距離を用いる手法に大別できる．中でもDPマッチングを用いる手法は，学習データを用いずに高い検索精度を得ることができるため，数多くの研究成果が報告されている．また，一般に音高差の音楽特徴量を量子化した系列に対してDPマッチングを行う手法が多いが，より検索精度を高めるために，音長や音高等の複数の音楽特徴量を統合してDPマッチングを行う手法も提案されている．更に，音符の削除や挿入に対してペナルティーを付加することが可能であり，学習データから適切なペナルティー値を求め，精度向上を図ることもできる．ただし，入力系列長 M, N に対するDPマッチングの計算量が $O(MN)$ となるため，大規模なデータに不向きといった欠点がある．一方，ユークリッド距離を用いる手法は，音高情報を音高推移特徴ベクトルと呼ばれる固定長ベクトルで表現し，ベクトル間の類似性をユークリッド距離で計算する．ユークリッド距離を用いることで，ユークリッド空間上のデータに対する索引化技術が適用できる．大規模なデータに対する検索速度の耐久性を考慮すると，索引化が適用できる手法が有効であると考えられる．そこで本稿では，索引化技術を適用できる，ユークリッド距離を用いる手法とDPマッチングを用いる手法を対象にして提案手法と比較検討を行う．

3. ハミングに対する採譜処理

ユーザからハミングされた音声データはWAVE形式のデータであるため，音響分析を行いハミングデータをMIDIデータ，即ち音符に準じたデータ形式に変換する必要がある．以下，この採譜処理の詳細について述べる．

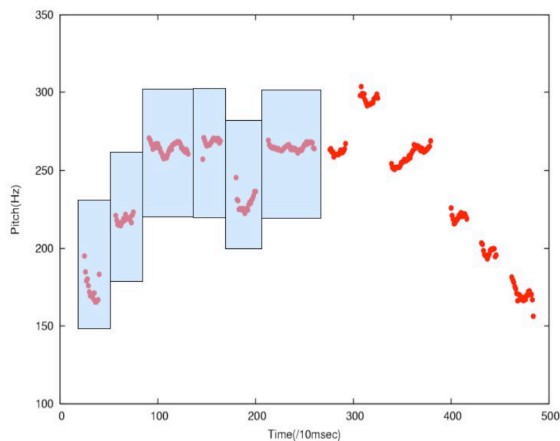


Fig.2 Fundamental frequencies for the humming data.

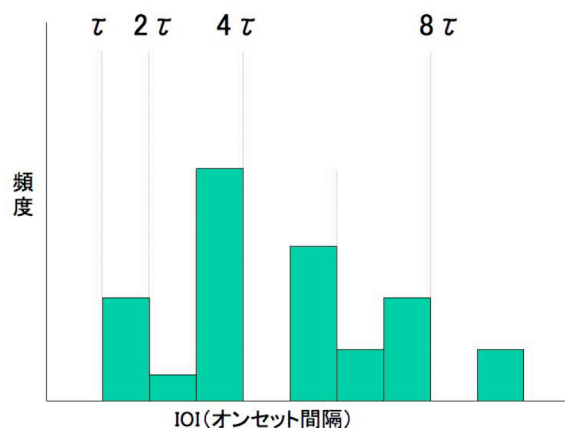


Fig.3 Distribution of the frequencies of IOI for the humming data.

まず、ユーザのハミングデータに対して図2に示すような時間経過による基本周波数の変化を分析する。図2の横軸は時間、縦軸は時間毎の基本周波数を表す。ユーザが入力するハミングは、“タッタッター”といったスタッカートを用いたハミングであるため、各音間に無音区間が存在する。そこで次ぎに、これらの無音区間に着目し、各音のオンセット間隔 (Inter-onset Interval: IOI) を検出する。図2においては、四角で囲まれた時間がオンセット間隔を表す。このオンセット間隔は、音の立上がりから次の音の立上がりまでの時刻と見なすことができる。

次に、ハミングデータ内の基準オンセット間隔 (以下、基準IOIと呼ぶ) を求める。この基準IOIは、ハミングデータ内の最短のオンセット間隔を表し、基準IOIを求めることで、他の音価 (音の長さ) を推定することになる。基準IOIを求めるためには、まず、図3に示すようなハミング内に存在するIOI頻度分布を計算する。ここで、ハミング内の音価は、ユーザー毎に揺れが生じるのは明らかである。同じ4分音符をハミングしてもユーザーによりその長さは異なる。しかしながら、4分音符と8分音符をハミングする長さの割合はどのユーザーも殆ど同じ (4分音符の約半分の長さで8分音符をハミングする)

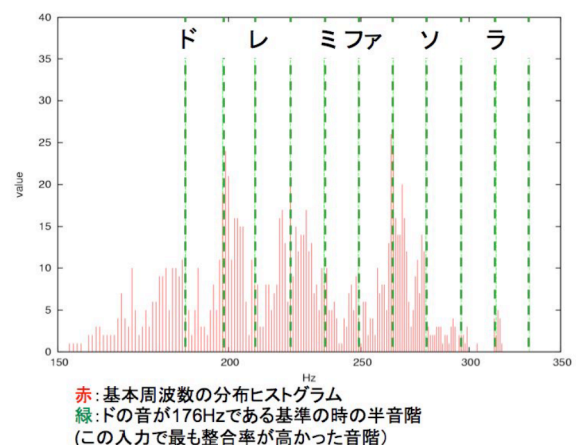


Fig.4 Distribution of the frequencies of fundamental frequencies for the humming data.

ではないかという点に着目し、本研究では音価テンプレートを用いて基準IOIを推定する。つまり、図3に示すように点線で示す音価テンプレートに対して、 τ の値を変化させてテンプレートを収縮する。そして、IOI頻度分布との整合性を計算し、整合度が最大になる τ を基準IOIとして決定する。基準IOIを決定した後が、テンプレートに従い、基準IOIを定数倍 (2倍、4倍、8倍) した値を他の音の音価とする。

音程の推定に関しては、まず、ハミングデータ内に存在する基本周波数の頻度分布を求める。そして図4に示すように、点線で示すような鍵盤テンプレートを収縮させながらスライドさせ、基本周波数頻度分布との整合度を計算する。その結果、整合度最大の周波数を基準として各音階周波数を推定する。図4の例では、“ド”の音が176Hzの際に整合度が最大になり、他の半音階毎の周波数を推定している。

4. EMDを用いたハミングによる類似音楽検索 4.1 ハミングによる音楽検索技術の課題

ユーザから入力されたハミングを採譜した結果得られるMIDI音楽データには以下のような特徴がある。

- ・特徴1：ユーザにより音程や音長に揺れが生じる。
- ・特徴2：冒頭やサビ以外の部分もハミングされる。
- ・特徴3：音符通り正確にはハミングされない。

上記のようなハミングの特徴から生じる課題と対応策についていかに列挙する。まず、特徴1から生じる課題として、音程や音長のズレを許容した検索が必要になる。そのため、絶対音程ではなく、相対的な音程により特徴量を生成する対応策が考えられる。次に、特徴2に対しては、曲の任意の箇所からの歌い出しにも検索できる必要があり、スライディングウィンドウ方式によりメロディ片に分割することで対応可能である。最後に、課題3については、音符の誤挿入、欠落、置換にも対応可能な検索アルゴリズムが必要になる。従来この点に関しては、DPマッチングが用いられていたが、本研究では、より頑健な検索が可能なEMDを用いる。

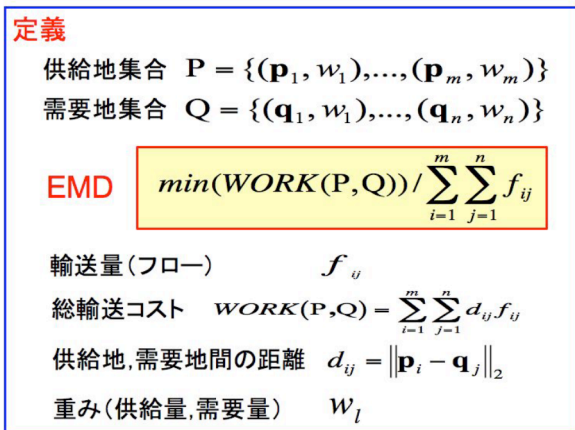


Fig.5 Definition and equations of EMD.

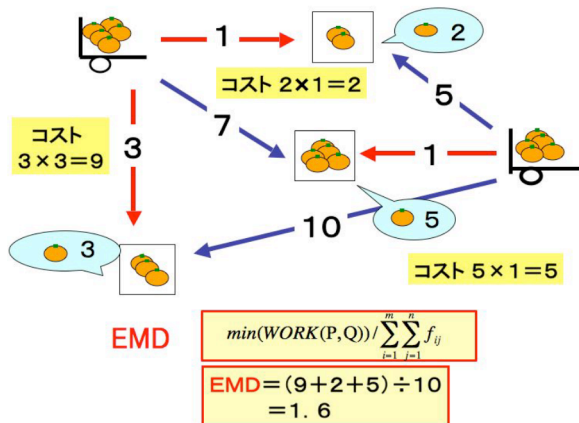


Fig.6 Example of calculation of EMD.

4.2 EMDとは

EMDは線形計画問題の一つである輸送問題の解に基づいて計算される。輸送問題とは一定の供給量を持つ複数の供給地と同じく一定の需要量を必要とする需要地を設定し、各供給地から需要地までの輸送コストが与えられた際に、需要地の需要を満たすよう供給地から需要地へ最小の輸送コストで荷物を輸送する輸送方法を探す問題である。以下、EMDの計算方法を説明する。

まず、 m 個の供給地を持つ供給地集合 P と n 個の需要地を持つ需要地集合 Q をそれぞれ図5のように表す。ここで、 p_i は i 番目の供給地を表す特徴ベクトル、 w_i は i 番目の供給地が有する供給量、 q_j は j 番目の需要地を表す特徴ベクトル、 w_j は j 番目の需要地が必要とする需要量を表す。各供給地 p_i と各需要地 q_j 間の単位輸送あたりの輸送コストを図5内の d_{ij} として定義する。一般に、輸送コストとして各ベクトル要素とのユークリッド距離を用いる。次に供給地・需要地間のすべての組み合わせを考慮し、総輸送コストを計算する。 P から Q への輸送量(フロー)を決定する以下の輸送問題の解を用いて計算する。任意の供給地・需要地の組み合わせによる総輸送コスト(WORK)は図5内の式で表すことができる。

ただし、総輸送コストを計算する場合、以下の制約条件を満たすものとする。

・制約条件1：供給地から需要地の一方向にしか輸送されない

$$f_{ij} \geq 0, (1 \leq i \leq m, 1 \leq j \leq n)$$

・制約条件2：供給地 i から供給できる容量は供給量 w_{pi} を超過しない

$$\sum_{j=1}^n f_{ij} \leq w_{pi}, (1 \leq i \leq m)$$

・制約条件3：需要地 j が受け取れる容量は需要量 w_{qj} 以下である

$$\sum_{i=1}^m f_{ij} \leq w_{qj}, (1 \leq j \leq n)$$

・制約条件4：供給地から移動できる最大総輸送量

$$\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min \left(\sum_{i=1}^m w_{pi}, \sum_{j=1}^n w_{qj} \right)$$

最終的にEMDは、上の輸送問題の最適値(総輸送コストの最小値)を総フローで割って求まる。

EMDの計算処理の例を図6に示す。図6において2台のトラックを供給地、3個の口を需要地とし、それぞれ図に示す資源が割り当てられている。尚、供給量、需要量の総数はどちらも10個で同数である。また、供給地から需要地への経路は矢印で示され、矢印上の値がその経路での輸送コストを表す。このような条件下で、実線矢印の輸送経路が選択された場合、図6の下方に示す計算式に従って値が求まる。

距離尺度にEMDを用いた画像検索手法では、色情報に基づいて画像を領域分割し、領域毎に領域の広さ、平均色、重心点、テクスチャ情報を抽出する。そして、画像内の領域数を供給地数、各領域の広さを各供給地が有する供給量とみなす。更に、各領域の平均色、重心点、テクスチャ情報を固定長のベクトルで表現し、各ベクトル間のユークリッド距離を輸送コストとみなす。そして、比較対象の二つの画像の一方を供給サイド、他方を需要サイドとみなして画像間の類似度をEMDによって計算する。EMDを適用した類似画像検索手法は、色合いのみでなく構図も考慮した画像を検索可能であり、高い検索精度が示されている。

4.3 EMDを用いた類似音楽検索手法

EMDを用いた類似画像検索が各領域の広さと各領域の画像的特徴の類似性を同じ距離尺度で判定できたように、類似音楽検索の距離尺度にEMDを適用すると、各音符の長さや各音符の音楽的特徴(出現時間、音高)の類似性を同じ距離尺度で判定でき、その結果、人間の感覚と同じように、全体の曲調が類似した曲を精度よく検索できるのではないかと我々は考えた。EMDを類似音楽検索手法に適用するにあたり、スライディング・ウインドウ方式⁽¹⁰⁾により分割されたメロディ片毎に音楽特徴量を作成する。これらのメロディ片毎に音楽特徴量を作成することによりインデキシングが可能になる。

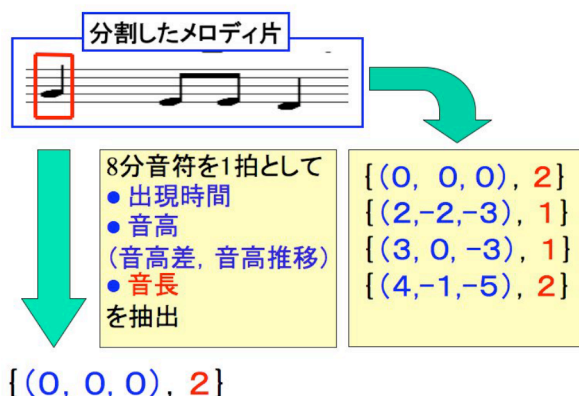


Fig.7 Example of musical signatures for EMD.

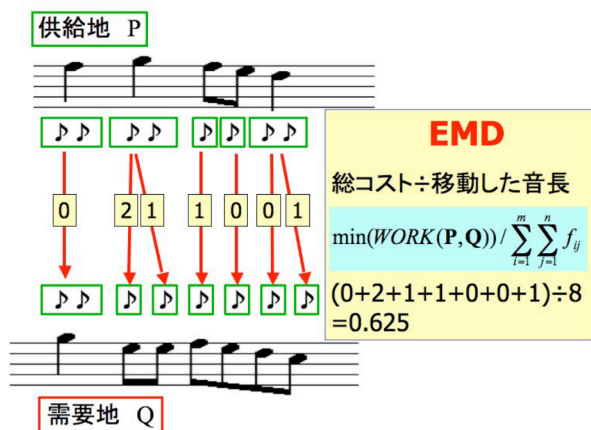


Fig.8 Example of calculation of EMD on the musical signatures.

また、メロディ片間の類似性はメロディ片内の音符を音長や音高が類似した他のメロディ片内の音符にマッピングするコストと考えられる。しかし、メロディ片内の音符数が一定でないため、時系列データである音符を分割してマッピングするコストの設定が必要である。このコストの計算にEMDを用いた場合、輸送問題における資源分配を音符の分割マッピングと同等に扱えるため、メロディ片内の音符数を供給地数、各音符の長さを各供給地が有する供給量とみなす。音符間の単位輸送コストを求めるための特徴量は、各音符の出現時間、前音との音高差、音高推移情報を固定長のベクトルで表現し、各ベクトル間のユークリッド距離を輸送コストとする。出現時間と音長は拍数で表現し、出現時間はウィンドウ内で当該音符が出現した拍数を表す。音高差は半音高い音を1、半音低い音を-1とする。

図7に8分音符を1拍とした場合の音楽特徴量の例を示す。メロディ片内の最初の音符に対する出現時間、音高に関する特徴量は0とする。音高差を特徴量に用いることで、ハミングにおけるキーの高さが異なっても音高差は同じになるので、ユーザ毎の音程の違いに対応することができる。図8に二つのメロディ片間に生じるEMDの計算例を示す。

このような音楽特徴量を採用することで、音程のずれに対しては、音符の音高情報が反映された輸送コストによりEMDが計算される。また、ハミングデータで頻出する音符の誤挿入に対しては、音符の出現時間といった時系列情報が反映された輸送コストに従って資源分配が行われ、EMDが計算される。特に、極端に音高の外れた音符（外れ音符）の誤挿入に対しては、DPマッチングを用いた場合、外れ音符の直前に位置する音符との音高差がそのままコストに反映されるため、外れ音符の有無が距離に大きな影響を与える。一方、EMDを用いた場合、外れ音符の周辺（前後）に位置する音楽特徴量が類似した複数の音符に資源分配のフローが作成され、最適なコストが計算されるため、外れ音符に対するノイズ軽減が期待できる。更に、輸送コストをユークリッド距離で計算する際に、音符の出現時間にパラメータ的な重みを与えることで、時間軸方向の制約を調整できる（パラメータを大きくすることで、時間軸方向の制約を強めた最適な資源分配方法が決定され、逆に弱くすれば、時間軸方向の制約が弱まる）。以上のように、音楽特徴量間の類似性をEMDを用いて判定することにより、ハミング検索で重要な課題になる音程のずれやリズムの違いに対して柔軟な検索が可能になる。

4.4 検索処理の高速化への改良

EMDを用いた検索処理では、検索速度が遅いことが実用化への問題となる。検索速度が遅い理由としては、以下の2点が挙げられる。

・距離計算回数が多い：

曲をメロディ片に分割しているため、1曲が複数のメロディ片に分割される。その結果、メロディ片数は膨大になり、すべてのメロディ片に対して距離計算を行うと、多大な計算時間が必要となる。

・時間計算量が大きい：

EMDの計算には一般にシンプレックス法が用いられるため、音符数に対して指数関数的に時間計算量が増加する。メロディ片内には音符数が多く（今回の実験では、2小節のメロディ片内での平均音符数は約14であった）、多大な計算時間が必要となる。

まず、「距離計算回数が多い」といった問題点に対しては、音楽データベースの索引化手法が適用できる。音高推移特徴ベクトルを用いた従来手法では、ユークリッド空間内のデータを索引化するSR-tree⁽²²⁾を採用している。一方、EMDはユークリッド空間では表現されないため、同様の索引化手法は適用できない。そこで本手法では、EMDが距離の3公理⁽²³⁾を満たす点に着目し、距離空間内のデータを索引化するVP-tree⁽²⁴⁾を採用し、検索速度の向上を図っている。

次に、「時間計算量が多い」といった問題点に対しては、EMDの下限値を用いた高速化アルゴリズムを導入している。純粋にEMDを計算すると音符数の増加に従って指数関数的に時間計算量が増加する。一方、EMDの下限値はユークリッド距離と同程度の計算コストで求められる。本研究では、EMDの下限値と順位キューを用いて高速に検索結果を得るアルゴリズムを提案した。本手法を用いることにより、真のEMD値を計算するコストを軽減でき、時間計算量を大幅に軽減できた。

Table1 Experimental result on the automatic music transcription.

	リズム認識率		階名認識率	
声質	市販採譜ソフト	階名認識率	市販採譜ソフト	階名認識率
鏡音レン	71.3%	96.6%	61.4%	89.4%
鏡音リン	67.5%	91.5%	65.8%	90.6%
平均	69.4%	94.0%	63.6%	90.0%

5. 評価

5.1 採譜処理に対する評価

(1) 実験条件

旋律、拍子、曲調の異なる7つの曲から8小節分の音符データを採用した。ハミングデータは音声合成ツールを用いて人工的に作成した。音声合成ツールにより、ハミングする歌手（声質）を2種類に変更し、それぞれの声質でハミングデータを作成した。作成したハミングデータ内には、スラーやノイズは存在しない。このような条件の下にて、本システムと市販採譜ソフト⁽²⁵⁾との認識率をリズム認識率、階名認識率といった尺度にて採譜処理の精度を評価した。

(2) 実験結果

採譜処理の実験結果を表1に示す。実験結果よりリズム認識率、階名認識率共に市販のソフトよりも認識精度が向上していることが分かる。これは、市販ソフトによる認識が閾値ベースの手法により認識しているのに対して、本手法では、テンプレートを収縮させるマッチングにより比率ベースの手法を用いていることに起因していると思われる。しかしながら、今回のデータにはスラーが入っておらず、“タッタッター”といったスタッカート気味のハミングに対しては精度が良いが、“ラララー”といった通常のハミングにはスラーが混入することが多い。今回の手法では、スタッカート気味のハミングを前提として、無音部分で音符を区切っていたが、スラーが混入した場合には音符間に無音部分が混入しない。そこで、今後、無音部分以外の音響特徴に着目してスラーが混入したハミングに対してもリズム認識率を向上させる手法を検討したい。また、今回採用したテンプレートはくし型テンプレートと呼ばれるものであるが、ハミングの仕方によってはテンプレートの形状を変化させた方が認識率が向上することが予測される。そこで、今後、形状を様々に変化させたテンプレートを作成し、個人のハミングの癖を許容する採譜手法について検討したい。

5.2 音楽検索処理に対する評価

(1) 実験条件

検索対象の音楽データベースとして、童謡、J-pop、演歌等のジャンルが含まれるカラオケ用MIDI音楽データ約4,300曲を使用した。これら市販のMIDI音楽データは、特定のチャンネルに主旋律が格納されているため、機械的に主旋律のデータのみを自動抽出した。その後、主旋律のデータに対して、スライディング・ウィンドウ方式を適用して、メロディ片を生成した。スライディン

グ・ウィンドウの条件としては、8分音符を1拍とし、ウィンドウ長16拍、スライド長4拍としてメロディ片を生成した。メロディ片内の各音符に対して8分音符の長さを1とした音長、出現時間を用いて音楽特徴量を作成し、音楽特徴量の索引化を行った。

検索入力には、男女15名が歌ったハミングを採用し、MIDI形式に変換した50曲を用いた。採譜方法としては、ハミングの長さが最低でもウィンドウ長を超える条件を義務付け、ハミングの際には、正確なハミングは要求せず、その曲を知っている人が聞いて分かるレベルの入力とした。そのため、入力ハミングにはリズム、音程のずれが生じた。また、検索結果の順位付けの方法は以下の手順に従った。まず、検索対象の音楽データと同じ条件下でハミングをメロディ片に分割し、分割したハミングメロディ片毎に上位K件の検索結果を得る。その後、すべてのハミングメロディ片の検索結果を統合し、類似度順にソートした結果を検索結果とした。よって、目的の曲と類似したハミングメロディ片が入力ハミング内に1つでも存在していれば、検索結果の上位に目的の曲がリストアップされる。検索結果を曲毎にマージする場合は、ハミングメロディ片に対する距離の逆数を類似度とし、曲毎の類似度の総数をソートする。尚、実験に用いた計算機は、DELL Precision 690, Xeon 3GHz×2, メモリは4Gであった。

(2) 実験結果

従来手法として、ユークリッド距離を用いた手法、DPマッチングを用いた手法を採用した。ユークリッド距離を用いた手法としては、メロディ片毎に音高推移ベクトルを生成し、ユークリッド距離でベクトル間の類似度を計算した。DPマッチングを用いた手法としては、音長と音高差の距離の総和を用いてDPマッチングを行い類似度を計算した。ただし、ハミング系列全体を用いるのが通常のDPマッチングであるが、文献(10)に従ってハミングメロディ片毎にデータベース内のメロディ片とのDPマッチングを行った。このような処置を行うのは、索引化を可能にするためだが、比較系列数が短くなるため、通常のDPマッチングと比べて検索精度が若干低下する。尚、検索結果の順位付けの方法は本手法と同様にした。図9に提案手法と従来手法との検索精度を示す。横軸は検索結果の順位、縦軸はその順位以内に正解のメロディ片が検出された曲数の割合（正解率）を表す。例えば、5位以内で80%の正解率の場合、入力ハミング50曲中40曲において5位以内に正解が含まれていたことを意味する。また、Kの値を20と40に変更した結果も示す。

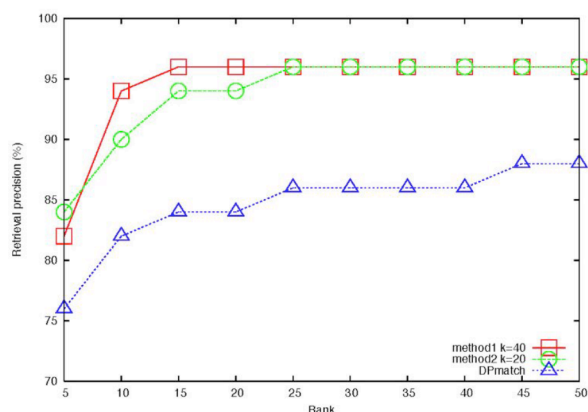


Fig.9 Experimental result of retrieval precisions.

実験結果からEMDを用いることで、従来手法より検索精度が向上したことが分かる。図9内には掲載されていないが、ユークリッド距離を用いた手法は、どの順位に対しても約55%程度であった。ユークリッド距離を用いた場合、特徴ベクトルは拍単位の音高情報から構成される。そのため、音高が類似した曲は検索できるが、リズムが異なった曲が検索されてしまい、精度の低下を招いていた。DPマッチングを用いた手法と比較すると、本手法の方が上位5位以内では約5%、上位10位以内では約12%向上している。DPマッチングよりも本手法の方が精度が良かったハミング片を調査したところ、ハミング内に極端に音高の外れた音符（外れ音符）を含んでいた。このような外れ音符が含まれていた場合、DPマッチングでは外れ音符と正解音符との差がそのままコストに反映され、正解データとの距離が大きくなる。入力系列数（音符数）が長ければ、少々の外れ音符が出現してもさほどコストに影響を与えないが、テンポがスローなハミングや今回のようなハミングメロディ片に対する系列の場合、音符数が少ないため、外れ音符の影響が大きくなったと考えられる。一方、提案手法を用いた場合は、時間軸方向の制約がDPマッチング程強くないため、外れ音符の周辺（前後）に位置する音楽特徴量が類似した複数の音符にフローが作成され、正解データとの距離の増加が抑えられたと考えられる。

次に検索速度について、ハミング1曲あたりの平均検索時間を各手法について測定した。尚、ユークリッド距離を用いる手法については、SR-tree、本手法についてはVP-treeによる索引化手法を適用した。その結果、ユークリッド距離を用いる手法は約0.2秒、DPマッチングを用いる手法は約8秒であったのに対して、提案手法は約0.7秒であり、高速な検索処理が実現できた。特に本手法において、EMD下限値による高速化アルゴリズムを適用せずに、純粋なEMD値を計算した場合には、検索時間は約15秒であった。このようにEMD下限値による高速化アルゴリズムを適用することで約20倍の高速化が実現でき、システム全体の時間計算量を大幅に軽減することに成功した。

6. まとめ

本稿では、ハミングに類似した音楽を検索するシステムについて紹介した。本システム内では、ハミングデータをMIDI音楽データに変換する採譜手法、距離尺度としてEMDを用いたハミングによる類似音楽検索手法、及びEMD下限値と順位キューを利用した高速化アルゴリズム“fastEMD”が用いられている。実際のハミングデータを用いた評価実験では、ユークリッド距離を用いた従来手法に比べて検索精度を向上することができ、DPマッチングを用いた手法に比べて時間軸方向の制約を柔軟に調整できることを確認できた。また、EMDの下限値を用いた高速化アルゴリズムを導入することで、検索処理の高速化が実現できた。

今後の課題としては、大規模なデータベースに対しては、検索処理の並列化が必須の課題になると考えられるので、新たな索引化手法の考案、データベースの冗長性の削減等に対して取り組みたい。また、より大量のハミングデータと比較し、本手法の頑健性を検証する予定である。

参考文献

- 1) 後藤真孝, 平田圭二: 音楽情報処理の最近の研究, 日本音響学会誌, Vol.60, No.11, pp.675-681 (2004).
- 2) 帆足啓一郎, 上月勝博, 菅谷史昭: 楽曲配信サービスを支える音楽情報検索技術, 電子情報通信学会誌, Vol.88, No.7, pp.529-534 (2005).
- 3) 武田晴登, 篠田浩一, 嵯峨山茂樹: リズムベクトルを用いたリズム認識, 情報処理学会音楽情報科学研究会資料, MUS-46-4, pp.23-28 (2002).
- 4) 池谷直紀, 服部正典, 大須賀昭彦: リズム入力による音楽検索方式「タタタタップ」, 第3回情報科学技術フォーラム, No.G-021, pp.391-393 (2004).
- 5) Dannenberg, R. B., Hu, N.: Understanding Search Performance in Query-By-Humming Systems, Proc. of 5th International Symposium on Music Information Retrieval, pp.232-237 (2004).
- 6) Colin, M., William, B.: Johnny Can't Sing: A Comprehensive Error Model for Sung Music Queries, Proc. of 3rd International Symposium on Music Information Retrieval, pp.124-132 (2002).
- 7) Jyh-Shing, R.J., Hsu, C.L., Lee, H.R.: Continuous HMM and Its Enhancement for Singing/Humming Query Retrieval, Proc. of 6th International Symposium on Music Information Retrieval, pp.546-551 (2005).
- 8) Ito, A., Heo, S.P., Suzuki, M., Makino, S.: Comparison of Features for DP-Matching Based Query-By-Humming System, Proc. of 5th International Symposium on Music Information Retrieval, pp.297-302 (2004).
- 9) Parker, C.: Applications of Binary Classification and Adaptive Boosting to the Query-By-Humming Problem, Proc. of 6th International Symposium on Music Information Retrieval, pp.245-251 (2005).
- 10) Steffen, P.: CubyHum: A Fully Operational Query By Humming System, Proc. of 3rd International Symposium on Music Information Retrieval, pp.187-196 (2002).
- 11) Sonoda, T., Muraoka, Y.: A WWW-based Melody-Retrieval System - An Indexing Method for A Large Melody Database, Proc. of ICMC 2000, pp. 170-173 (2000).

- 12) 園田智也, 後藤真孝, 村岡洋一: WWW上での歌声による曲検索システム, 電子情報通信学会論文誌, Vol.J82-D-II, No.4, pp.721-731 (1999).
- 13) 西村拓一, 滝田順子, 後藤真孝, 岡隆一: 類似メロディー区間検出による音楽時系列検索の高速化, 情報処理学会音楽情報科学研究会資料, MUS-39-10, pp.63-70 (2001).
- 14) 許盛弼, 鈴木基之, 伊藤彰則, 牧野正三: 複数の音高候補値を用いた楽曲検索システムの構築, 情報処理学会音楽情報科学研究会資料, MUS-49-15, pp.85-90 (2003).
- 15) 小杉尚子, 小島明, 片岡良治, 串間和彦: 大規模音楽データベースのハミング検索システム, 情報処理学会論文誌, Vol.43, No.2, pp.287-298 (2002).
- 16) 小杉尚子, 櫻井保志, 山室雅司, 串間和彦: Sound Compass: ハミングによる音楽検索システム, 情報処理学会論文誌, Vol.45, No.1, pp.333-345 (2004).
- 17) Dannenberg, R. B., William, P. B., George, T., Colin, M., Ning, H., Bryan, P.: The MUSART Testbed for Query-By-Humming Evaluation, Proc. of 4th International Symposium on Music Information Retrieval, pp.41-50 (2003).
- 18) 熊本忠彦, 太田公子: 印象に基づく楽曲検索研究のための印象表現の収集, 情報処理学会論文誌, Vol.43, No.10, pp.3231-3234 (2002).
- 19) Hoashi, K., Matsumoto, K. and Inoue, N.: Personalization of user profiles for content-based music retrieval based on relevance feedback, Proc. of 11-th ACM International Conference on Multimedia, pp.110-119 (2003).
- 20) Grachten, M., Arcos, J.L., Mantaras, R.L.: Melodic Similarity: Looking for a Good Abstraction Level, Proc. of 5th International Symposium on Music Information Retrieval, pp.210-215 (2004).
- 21) Rubner, Y., Tomasi, C and Guibas, L.-J.: The earth mover's distance, multi-dimensional scaling, and color-based image retrieval, Proc. of the ARPA Image Understanding Workshop, pp.661-668 (1999).
- 22) 片山紀生, 佐藤真一: SR-tree: 高次元点データに対する最近接検索のためのインデックス構造の提案, 電子情報通信学会論文誌, Vol.J80-D-I, No.8, pp.703-717 (1997).
- 23) Yianilos, P.N.: Data structures and algorithms for nearest neighbor search in general metric spaces, Proc. of the ACM-SIAM SODA'93, pp.311-321 (1993).
- 24) Fu, A.W.-C., Chan, P.M.S., Cheung, Y.-L. and Moon, Y.S.: Dynamic vp-tree indexing for n-nearest neighbor search given pair-wise distances, VLDB Journal, pp.2-8 (2000).
- 25) 鼻歌ミュージシャン2: 株式会社メディア・ナビゲーション.